

# Integrated Design and Simulation for Millimeter-Wave Antenna Systems<sup>12</sup>

Tom Cwik, Daniel S. Katz and Frank J. Villegas  
Jet Propulsion Laboratory  
California Institute of Technology  
Pasadena, CA 91109  
818-354-4386  
cwik@jpl.nasa.gov

**Abstract**—Several instruments operating in the microwave and millimeter-wave bands are to be developed over the next several years at either JPL or JPL in conjunction with various other companies and laboratories. The design and development of these instruments requires an environment that can produce a microwave or millimeter-wave optics design, and can assess the sensitivity of key design criteria (beamwidth, gain, sidelobe levels, etc.) to thermal and mechanical operating environments. An integrated design tool has been developed to carry out the design and analysis using software building blocks from the computer-aided design, thermal, structural and electromagnetic analysis fields. The capability to simultaneously assess the effects of design parameter variation resulting from thermal and structural loads can reduce design and validation cost and generally lead to more optimal designs, hence higher performing instruments.

In this paper the development and application of MODTool (Millimeter-wave Optics Design), a design tool that efficiently integrates existing millimeter-wave optics design software with a solid body modeler and thermal/structural analysis packages, will be discussed. The design tool is also directly useful over other portions of the spectrum, though thermal or dynamical loads may have less influence on antenna patterns at the longer wavelengths. Under a common interface, interactions between the various components of a design can be efficiently evaluated and optimized. One key component is the use of physical optics analysis software for antenna pattern analysis. This software has been ported to various platforms including distributed memory parallel supercomputers to allow rapid turn-around for electrically large designs.

## TABLE OF CONTENTS

1. INTRODUCTION
2. INTEGRATED DESIGN
3. HIGH-FIDELITY MODELS
4. THE MODTOOL ENVIRONMENT

5. EXAMPLE BY TUTORIAL
6. DISCUSSION
7. ACKNOWLEDGMENTS

## 1. INTRODUCTION

The process of designing and analyzing a multiple-reflector millimeter-wave instrument has traditionally been time-intensive, requiring large amounts of both computational and human time. Beginning with a conceptual paper design and ending with a hardware prototype, tools from a range of disciplines are necessary to produce a microwave or millimeter-wave optics telescope, and to assess sensitivity of key design criteria (beamwidth, gain, sidelobe levels, etc.) to thermal and mechanical operating environments. This paper describes an integrated design tool to carry out the design and analysis using software building blocks from the computer-aided design, thermal, structural and electromagnetic analysis fields. The software environment is termed MODTool, for Millimeter-wave Optics Design Tool. It has been found that the capability to simultaneously assess the effects of design parameter variation resulting from thermal and structural loads can reduce design and validation cost and generally lead to more optimal designs, hence higher performing instruments. A significant component of this process is the ability to transparently link these loads, as well as differences in manufactured components relative to their nominal design, back into the design environment. The ability to assess the performance degradation due to differences from nominal designs is integral to the MODTool environment.

The motivation for this specific design environment will be introduced along with some specifics of the design software pieces that are not commercially available. This will be followed by a description of the software environment underlying MODTool and will conclude with an example of its use.

<sup>1</sup> 0-7803-6599-2/01/\$10.00 © 2001 IEEE

<sup>2</sup> Updated November 8, 2000

## 2. INTEGRATED DESIGN

The goal of this development was to create an environment for design, analysis and simulation of millimeter and microwave instruments that can be used both in proposal stages and in the project design phase. MODTool was developed for quick design tradeoffs where instrument performance is coarsely predicted at the proposal stage as well as in latter design stages where accurate models and predictions are necessary before prototypes are built. This goal led to the development of user interfaces specific to the class of instruments being developed, well defined parameterizations of the instrument that tie different disciplines together (electromagnetic, structural, thermal, etc.), the use of high fidelity software tools, and the use of high performance computers to provide fast turn-around in specific design areas. Four key attributes characterize the MODTool design environment

1. High-fidelity modeling
2. A common digital geometry
3. Collaborative environment
4. Fast design turn-around

*High-fidelity modeling* – the first key characterization of the MODTool design environment is the use of engineering design software that predicts to high accuracy the instruments' performance. A wide variety of software tools have been integrated into MODTool. One important tool in this suite accurately calculates the complete antenna pattern of the system using physical optics diffraction software. This software is a JPL-written analysis package that can require long execution times for high frequency instruments consisting of multiple mirrors. MODTool has also provided interfaces to a set of commercial software packages that are used in instrument modeling. Because millimeter-wave telescopes have sets of mirrors that can be described similarly to purely optical systems, an interface to the ZEMAX optical modeling software has been created that allows ZEMAX to be used as a graphical interface for building mirrors in well-defined instrument coordinate systems. Prescription files from ZEMAX can be used to define input mirror files for the physical optics diffraction calculations. Important to the analysis of an instrument is the degradation of performance when subject to thermal and dynamical loads. For a given spacecraft trajectory and operating environment stresses can deform the telescope mirrors, altering the gain or sidelobe levels with resultant degradation of the signal-to-noise ratio of the instrument. MODTool imports load files that define surface distortions due to these loads. Interfaces to NASTRAN structural models using the FEMAP package are available, and thermal and dynamical analysis resulting in load files have been generated using the commercially available Mechanical Desktop, Thermal Desktop and Adams packages.

*Common digital geometry* – it was necessary for all design software to work with a common model of the instrument. Two aspects of a common digital geometry were recognized as being important. The first was the direct transfer of data between commercial software packages, as well as between the various commercial software packages and the internal JPL-developed software. This was accomplished using standard graphical and data formats available in the commercial software and by writing special translators that take the output of some of the commercial packages (such as ZEMAX) and create appropriate data files that contain geometry descriptions for the physical optics software. The second aspect of a common digital geometry was the definition of discretized models used for structural, thermal and the diffraction calculations. Typically, a diffraction calculation requires a few sample points per electrical wavelength over the reflector surface. The surface is discretized into a faceted surface at this density and can result in thousands to millions of facets for electrically large dishes. Calculations are then performed over the facets. Thermal calculations using a finite element analysis typically require one or two orders of magnitude fewer discretized sample points. Similarly, structural calculations require many fewer sample points than the thermal models depending on the model's fidelity. After initial use of MODTool in a design setting it was decided that the common digital geometry defining a design would not be a discretized set of points, but would be a parameterized model that could latter be discretized as needed. Each analysis software package would be able to read the common parameterized model and discretize it to the proper density for calculation. When transferring a load file containing structural distortions to the physical optics diffraction code, interpolation algorithms are used to generate surface data at the proper discretization.

*Collaborative environment* – having a complete design and analysis environment allows simulations of the instrument and spacecraft under its operating conditions. MODTool has been integrated into the Payload Development Team, a design center at JPL for instrument and payload development at the proposal and early design phases [1]. The instrument model is integrated with spacecraft orbital analysis software (SOAP) that, among other things, returns the instrument location relative to the Sun, as well as the resultant thermal load that is used in a thermal and structural analysis. The integrated software set provides a collaborative environment where various analysts, instrument principal investigators and designers can gather to simulate system performance to high fidelity at the proposal and early design stages.

*Fast turn-around* – in a collaborative environment it is desirable for the instrument's system performance to be relatively quickly calculated. Perhaps more precisely it is important that any one part of the design or analysis is not especially slow and out of synch with the other pieces of a

simulation. MODTool allows the interaction of a range of tools most of which are efficiently executed on workstations or personal computers. Nevertheless, the physical optics software can take hours of processing time for electrically large antenna mirrors. To greatly reduce execution time for this part of the MODTool simulation, the physical optics code was ported to high performance cluster computers [2].

Use of the cluster computer was made transparent to MODTool users via automatic file transfer and translation, and all parallelization aspects of the physical optics code and use of the cluster are similarly transparent. Another important aspect to fast turn-around is the use of a common digital geometry to describe the instrument and a common file system that holds results of the various simulations. The data is all local to the MODTool environment and easily available for all aspects of the simulation.

### 3. HIGH FIDELITY MODELS

All software integrated into MODTool provides high-fidelity predictions of instrument performance. Orbital trajectories, thermal models, structural and dynamic models are all precise predictions with levels of accuracy depending on the detail of modeling. This detail includes material parameters, design geometry, bandwidths of operation among others, with documentation available with the specific software tool. Being an internally developed package the diffraction calculation will be outlined in greater detail.

#### 3.1 Physical Optics Approximation

In reflector antenna systems, the mirror conducting surfaces are usually large with respect to the wavelength  $\lambda$ . In addition, we can also deem the reflector surface *locally flat* as a result of its low-curvature shape. This second condition is of crucial importance because we can impose an approximation known as the *physical optics (PO) approximation*. On a flat, perfect electric conductor of infinite extent the induced current density is

$$\mathbf{J}_s = \hat{n} \times (\mathbf{H}^i + \mathbf{H}^s) = 2\hat{n} \times \mathbf{H}^i, \quad (3.1)$$

where  $\mathbf{H}^{i,s}$  indicates the incident and scattered magnetic field, respectively. The relation shown in Eq. (3.1) holds because the incident and scattered tangential field components have equal amplitude and phase. On an arbitrary conducting body, this definition does not hold. However, when a perfect electric conductor is electrically large and has a rather large radius of curvature, Eq. (3.1) can still be used, if only in an approximate sense. Thus, the physical optics approximation consists of using

$$\mathbf{J}_{PO} \approx 2\hat{n} \times \mathbf{H}^i, \quad (3.2)$$

as the total current induced on the surface of the reflector.

Note that the physical optics current of Eq. (3.2) is an equivalent current that coincides with the surface of the reflector, which has been removed and replaced by homogeneous media.

MODTool's physical optics analysis engine consists of using the physical optics approximation of Equation (3.2) to construct the currents on the mirror surfaces. Initially, the incident magnetic field is that due to the source, which in many cases may be an analytical model, i.e., that of a horn with a pre-defined taper. In subsequent steps, the physical optics current on the subreflector is used to generate the magnetic field incident on the main reflector. This is done using the diffraction integral

$$\mathbf{H}(\mathbf{r}) = \int_S \mathbf{J}_{PO}(\mathbf{r}') \times \nabla' G(\mathbf{r}, \mathbf{r}') dS', \quad (3.3)$$

where  $G(\mathbf{r}, \mathbf{r}') = e^{-jk_o R} / (4\pi R)$  is the free space Green function,  $k_o$  the free space wavenumber, and  $R = |\mathbf{r} - \mathbf{r}'|$  is the distance between source and observation points. The free space Green function can be used since by definition the physical optics current radiates in homogeneous matter (air in our case). Since physical optics is used on both the sub- and main reflectors in this analysis scheme, it is usually denoted as PO/PO. Numerically, the PO/PO algorithm used in MODTool consists of the following steps [3]:

1. The mirror surfaces are discretized into planar triangular facets.
2. Over each facet, the physical optics current is assumed *constant* (magnitude & phase).
3. Compute the physical optics currents on the first mirror's facets due to the incident magnetic field of the source.
4. Compute the main reflector physical optics current on the  $n^{\text{th}}$  facet due to the magnetic field radiated by the entire sub-reflector, i.e.,

$$\mathbf{J}_n^{\text{main}} \approx 2\hat{n} \times \mathbf{H}^i[\mathbf{J}_{po}^{\text{sub}}]; \quad n = 1, 2, \dots, N, \quad (3.4a)$$

where

$$\mathbf{H}^i[\mathbf{J}_{po}^{\text{sub}}] \approx \frac{1}{4\pi} \sum_{m=1}^M \left( jk_o + \frac{1}{R_{mn}} \right) \frac{e^{-jk_o R_{mn}}}{R_{mn}} \hat{R}_{mn} \times \mathbf{J}_m^{\text{sub}} \Delta_m \quad (3.4b)$$

and  $R_{mn}$  is the distance from the  $m^{\text{th}}$  sub-reflector facet to the  $n^{\text{th}}$  facet of the main reflector.

5. Repeat step 4 for all mirror pairs in ray-optics sequence.
6. Compute the final far field quantity by taking the contribution from all  $N$  main reflector triangles.

We should note that parallelization of this algorithm is



straightforward. In particular, the  $M$  sub-reflector facets and  $N$  main reflector facets are distributed evenly over all available processors. The computation of the  $M$  sub-reflector currents is done in parallel. Subsequently, the sub-reflector currents are redistributed such that all processors contain all  $M$  facet currents. Next, the  $N$  main reflector currents are computed in parallel consistent with the facet distribution over the processors. Finally, at each observation point in the far field, the contribution from the  $N$  main reflector currents is integrated on each processor, followed by a global summation over all processors.

#### 4. THE MODTOOL SOFTWARE ARCHITECTURE

MODTool was designed to:

1. Be a single graphic user interface (GUI), usable on Windows, Mac, and Unix platforms, with minimal downloading and building of software on these machines
2. Use common storage for files from CAD, meshing, thermal, and structural packages
3. Allow use of parallel computers for PO analysis while hiding this from the user as much as possible.

These requirements were met by creating a distributed client-server tool, with the client being written in *Tcl/Tk*.

##### 4.1. The MODTool Client

The client is distributed as a single script. As of MODTool 1.13, the script is approximately 340 Kbytes in length. This script is also a pure ASCII file, and it may be easily downloaded to any type of machine from an http or ftp server. In order to execute the script, the user must also acquire the *Tcl/Tk* binary package, which is already preloaded on many Unix (specifically Linux) machines. On other machines, we can take advantage of the *Tcl/Tk* community's effort to make binaries easy to download and build. The web site:

<http://dev.scriptics.com/resource/software/ports/>

contains a set of links to *Tcl/Tk* binaries which are

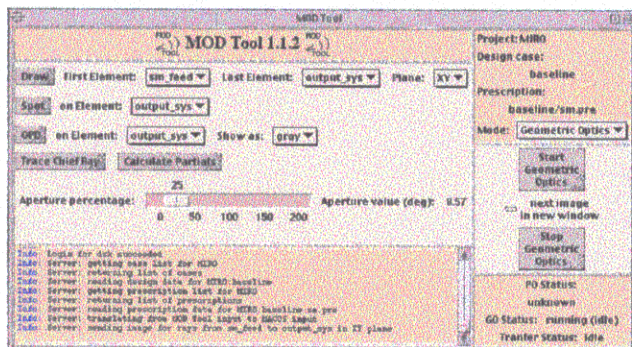


Figure 1. The main MODTool client window.

precompiled for many machines (10 as of 31 Oct. 2000). These are generally packaged in the simplest method for each machine (.sea.hqx for Mac, .exe for Windows, package format for Solaris, etc.) The MODTool user must download and install the correct *Tcl/Tk* the first time MODTool is downloaded and used. When MODTool changes, the *Tcl/Tk* binaries do not need to be re-loaded, just the new MODTool script.

Figures 1 and 2 show sample snapshots of the MODTool client. Figure 1 is the main MODTool client window. The user has selected a design case and a prescription for the instrument under design, and in Geometric Optics mode, has run a ray-trace, using the Draw button in the upper-left corner of the window. Figure 2 shows the window that popped up as the result of that command, and displays the rays going from the element *sm\_feed* to the element *output\_sys* in the YZ plane. These choices were made using the pulldown menus on the same line as the Draw button in the client. Other Geometric Optics analyses are possible, as shown by the other lines in the client, each with a button possibly followed some selections that need to be made for that type of analysis. The style of the buttons, pulldown menus, and the other widgets show in the client window, as well as the style of the window itself are specific to the client machine being used. Other machines would look slightly different, but the relative layout of the widgets

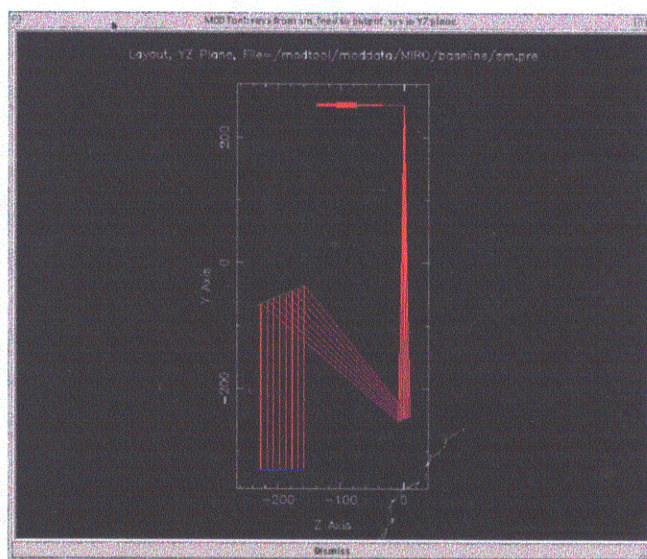


Figure 2. An example of a Geometric Optics Output window.

would be the same, and each widget would look like other widgets of that type from other applications on that machine.

##### 4.2 The MODTool Server

The MODTool server is written in *Expect*, an extension to *Tcl* that is commonly used to allow a computer to play the role of a human to automate testing of software. We take

advantage of this ability to allow the server to interact with many software packages that were designed for human interaction. These include: MACOS, a geometric optics package developed at JPL [4]; POPO, a Physical Optics code also written at JPL; and various scripts used for uploading information from the CAD, meshing, thermal, and structural packages. Because the server is running on a single machine and it contains the data files from the other disciplines, these files are available to all MODTool users in a simple manner through the client. They are stored in a logical structure that will be described later in this section.

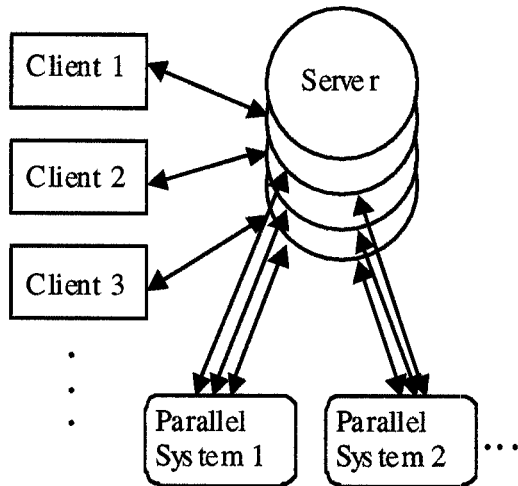


Figure 3. The relationship between the clients, server, and the parallel systems used for the PO analysis.

Because the server is written in *Expect*, it can communicate with other computers, such as parallel systems or supercomputers as if the client software user was directly communicating with them, as shown in Figure 3. This makes the use of the parallel computers relatively simple from the point of view of the server, since the user must already have an account on the parallel system, files may be stored in the user's directories, and accounting does not need be done by the server. We use *ssh* and *scp* to communicate between the server and the parallel systems, where *scp* is used to move files, and *ssh* is used to issue commands. For a given physical optics analysis, the server creates the input files for the parallel system just as it would if the PO code was running on the server. When the PO code on the server is started, the server instead transfers the files to the parallel system and starts the PO code on that system. Status files output by the PO code are similarly used; rather than reading the files directly from the server, they are first transferred from the parallel system, then read as usual. The first time any parallel system is used during any MODTool session, the client asks the user for a username/password pair. It transfers these to the server, and attempts to use them for the action being performed. If they work, they are saved during that MODTool session and used for each action involving that parallel system. If

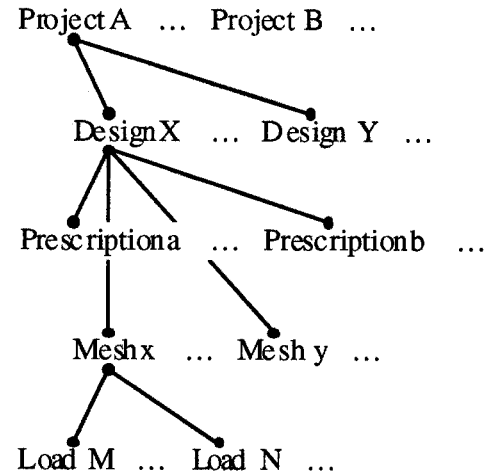


Figure 4. The MODTool server structure for file storage.

they don't work, the user is prompted to try again. When the session ends, the memory used to store the username and password is cleared.

As shown in Figure 4, the server is not a single process. In the configuration used at JPL, there is a single server process that is started when the Sun that hosts the server is rebooted. This server initializes a number of basic things, and then waits for connections from clients. When a client connects, the server uses the *Expect fork* routine to make a copy of itself. Now two copies of the server exist, each with connection to the client, and each waiting for additional clients to connect. The new process stops waiting for new clients and only works with the existing client, while the old server process breaks its connection to the client and just waits for more clients to connect. When another client connects, the same process happens again. Thus, at any given time there is one server process waiting for connections for clients, and the same number of server processes as connected clients exist. Each of the new server processes has its own memory, and is completely independent of the other servers, except for shared disk storage. These servers exist as long as their client is connected; when a client disconnects (possibly when it exits) the corresponding server is killed. In general, the term *server* is used for the server process corresponding with a given client, as well as for the machine on which the server processes exist.

The server is used to store a number of items that are either useful to multiple MODTool users or allow a single user to access the same data from multiple work areas. This includes the MODTool username/password pairs, used to permit access to the MODTool server from a client. Passwords are stored using the minimally-secure UNIX crypt protocol. The server also keeps a working area for each user. This allows the user to start a PO analysis, exit the client, and have the data still available when they start a



new session. Finally, the server stores the data associated with various projects, as shown in Figure 4. The highest level in this 'database' is the project level. Next is the design case level, followed by the mesh level, and then the load level. A project is a set of designs for a given instrument. A design case is a particular design. For each design, a number of prescriptions may exist (a prescription is an ordered list of the elements involved in an electromagnetic analysis of an instrument). A mesh is one example of many possible meshes for that particular design, and a load is one particular set of deformations to that mesh, based on thermal and/or structural analysis. An example of this is shown for an example instrument (A), a mesh (x = initial) was created for the initial design (X = initial), and two loads were analyzed, the first load (M = load one) and a second load (N load two). Later, other designs (Y ...) were created based on the results of the analysis, and for each design, various meshes and load were made.

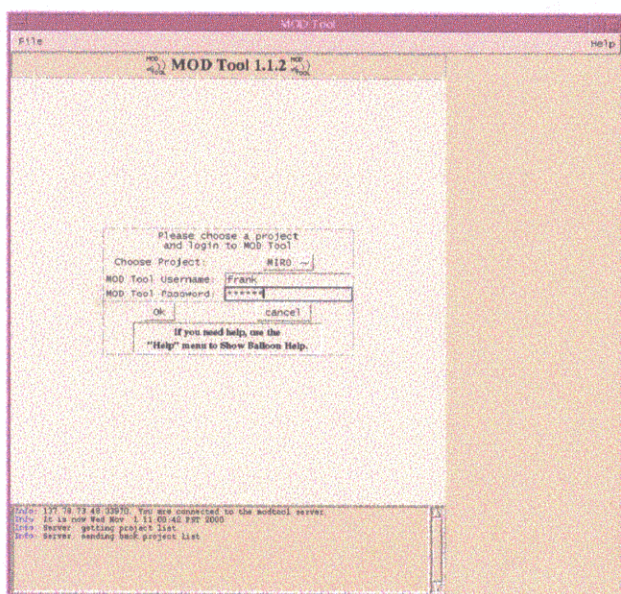


Figure 5. MODTool's login screen used to securely log on to a given project on the server.

## 5. EXAMPLE BY TUTORIAL

In this section, we will present a 'guided tour' of MODTool by way of a short tutorial. Our goal is to illustrate the ease with which an antenna engineer can use MODTool to analyze the radiation characteristics of multiple-reflector systems. Both nominal designs as well as those thermally (structurally) impacted by the operating environment can be evaluated in a real-time concurrent design setting. As mentioned in previous sections, this is made possible primarily through the use of a distributed parallel processing architecture, which makes the analysis of mirrors on the order of 500,000 tenable. As a practical

example, we will illustrate the efficacy of MODTool using MIRO, the Microwave Instrument for the Rosetta Orbiter [5]. This instrument, designed at JPL, contains a multiple-mirror antenna system operating in both the millimeter (190 GHz) and sub-millimeter (564 GHz) wavelength bands. The MIRO instrument will be placed on a spacecraft that will follow a comet as it travels from 3.2 AU to 1.1 AU. As it moves closer to the Sun, thermal effects may alter the topology of the mirror surfaces in such a way that performance is degraded. Determining the extent of these effects and compensating for them ahead of time is obviously crucial for a successful mission. In what follows, we will proceed sequentially through the various stages of the MODTool analysis of MIRO.

Figure 5 illustrates the MODTool login screen. This initial screen requires the user to choose a design project to work on and provide a username and password. The client communicates this data to the server for verification purposes, and subsequently accesses the design database to retrieve the relevant model files. At this point, the main MODTool screen comes up, allowing the user to work in one of five modes.

### 5.1 Design Mode

This mode allows the user to load, modify, and save a design. A design is defined as a set of physical optical elements (mirrors) along with two non-physical elements, namely, the feed and pattern coordinate system locations. Mirrors are represented as segments of conic sections obtained using a properly defined oval cutting cylinder. The canonical set of conics includes flat plates, paraboloids, ellipsoids, and hyperboloids. A point on its aperture and an additional point indicating the radiation direction define the feed location. The aperture radii must also be provided. The pattern coordinate system is similarly defined by an origin and a radiation direction. The design information is presented in tabular and graphical form in the main window, as shown in Fig. 6 for the MIRO case.

The table provides the coordinates of points that uniquely define the object (e.g., for a paraboloid, these would include the vertex, focus, and two points that define the axis of the cutting cylinder, in addition to the cylinder radius). Other information that might be useful to the designer is shown as well. The graphical view window (Figure. 6) depicts a planar cross-section of the mirror arrangement, an indispensable visualization tool for the designer, and also includes some primitive editing capabilities. The tabular and graphical windows are linked, and thus changes in either one are reflected in the other instantaneously (if a point on the drawing is moved, the corresponding table entries are updated, and vice versa). An additional useful feature of the graphics frame in the design mode is the user has the ability to work in any Cartesian plane (e.g., the xy-plane at a given value of z). Finally, because of the



client/server environment, designs are examined and modified by the GUI client, while the data is loaded from and saved to the server's database. This maintains all project files in a central location for efficient maintenance, such as scheduled backups.

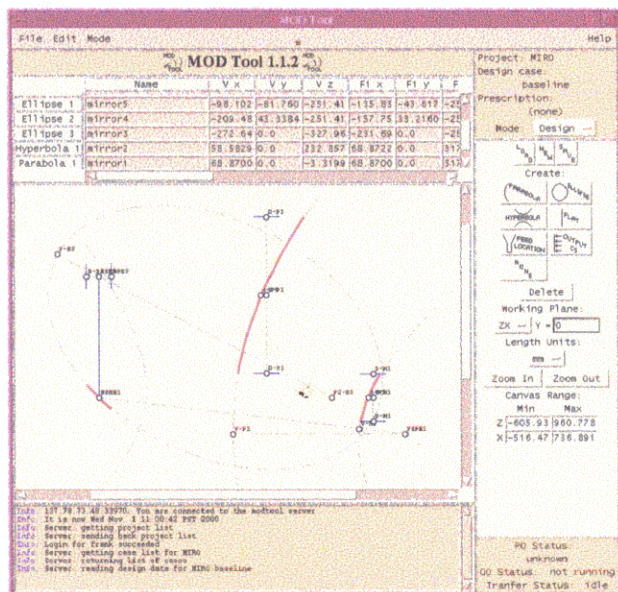


Figure 6. MODTool's Design Mode window. The MIRO mirror case is shown. In this planar view (zx-plane at  $y = 0$ ) only 3 of 5 mirrors are visible, the others residing on offset planes. Red segments indicate actual mirror surfaces, obtained using appropriate cutting cylinders on the canonical conic sections, shown in gray.

### 5.2 Prescription Mode

The prescription mode allows the user to specify and modify the objects that will be used in a particular analysis. Essentially, it defines the optical path for the multiple mirror arrangement, including the feed and output (pattern) coordinate system. A feed file, which defines the feed horn in one of various acceptable forms, is also selected here. The user can choose any subset of objects from the current design and arrange them in a desired sequential fashion (in a ray-tracing sense), or load a pre-defined prescription. Individual objects may also be translated and/or rotated in order to perform a tolerance analysis. Once the set of optical components has been chosen and properly ordered, the user can save the prescription. An additional parameter—parts per wavelength—(ppw) is also specified here for each object, indicating the average electrical distance between nodes of the meshed surface topology. Again, as a result of the client/server implementation, prescriptions are examined and modified within the GUI client, but the data is loaded from and saved to the server's design database. Figure 7 illustrates a particular sub-millimeter wave prescription used for the MIRO project.

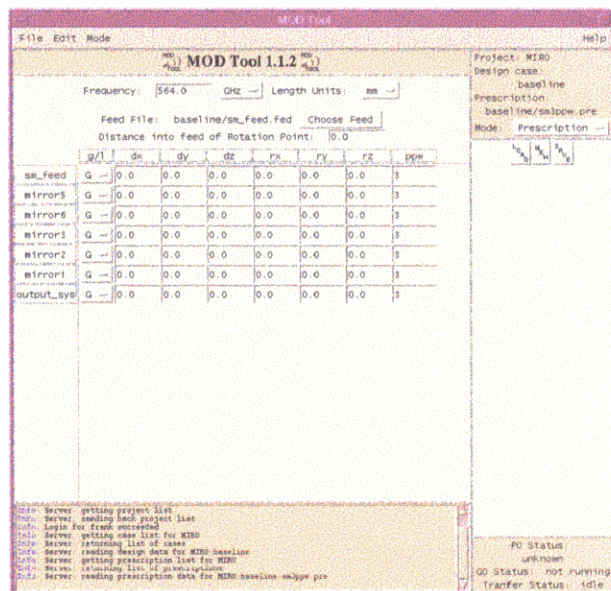


Figure 7. MODTool's Prescription Mode window showing a sub-millimeter (564 GHz) prescription for the MIRO instrument. The multiple-reflector system's optical path is comprised of five mirrors, in addition to the feed and output coordinate system.

Note that at this frequency (564 GHz), the wavelength is approximately 0.5 mm. MIRO's main parabolic reflector is on the order of 325mm in diameter, which is equivalent to  $650 \lambda$ . Its surface area is approximately  $330,000 \lambda^2$ . Based on the mesh density implied in Fig. 7 (3 points per wavelength – 3 ppw), the main mirror surface alone encompasses approximately 1,000,000 triangles. This is an amazingly large number of unknowns for any full-wave numerical scheme, as is the PO/PO algorithm employed by MODTool. A single run of this analysis on a Sun Ultra-1 takes more than 12hrs to complete, compared to a 15 minute run time on a cluster computer consisting of 32 compute nodes.

### 5.3 Physical Optics Analysis Mode

The PO/PO algorithm, described in a previous section, is a very accurate and efficient way to compute the PO currents on the mirror surfaces and in turn the fields radiated by each reflector. Parallel implementation is rather straightforward as a result of the simplicity of the algorithm. Figure 8 illustrates the Physical Optics Mode for MODTool. This mode allows the user to choose an appropriate server, set far field parameters, and launch PO runs. Depending on the electrical size of the mirrors, a supercomputer may be a prudent choice, as is the case with MIRO. At present, the PO code used at JPL has been ported to the Cray J90, HP SPP-2000 and Beowulf systems. Starting a PO job on a supercomputer involves: the server's conversion of the



design and prescription information into a format consistent with the PO code, the client obtaining a username and password for the supercomputer from the user, and the server's transferring of files to the supercomputer followed by a run initiation. As we've

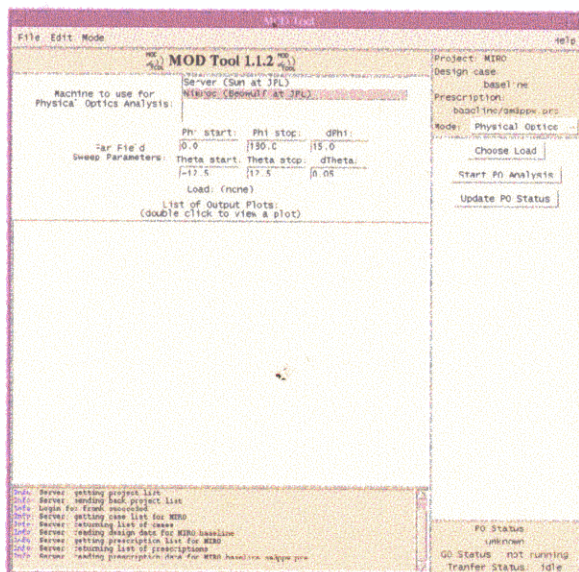


Figure 8. The Physical Optics Mode window. The server machine is chosen, and pattern calculation parameters are set. The user starts the PO analysis, and checks the run progress. Directivity patterns can be viewed from this screen, and data saved to text files.

alluded to, the PO code can also be used to analyze surfaces that have been structurally deformed due to thermal loading. This is accomplished by incorporating a mesh/load file combination that has been submitted (using the Submit Data Mode below) to the server. A MATLAB script is used to calculate the coefficients of a bi-polynomial representation of the surface distortions, and this information is subsequently transferred to the PO code on the supercomputer.

#### 5.4 Submit Data Mode

The structural engineer can use this mode to forward the mirror deformation information to the antenna designer in real time. Two types of data can be submitted: mesh and load files. Once the engineer has created a mesh, based either directly on the design or a CAD model that was built from the design, he uses this mode to submit it from the client to the server, which stores it in the relevant design database. MODTool currently uses FEMAP neutral files for meshes, which are unit-independent. As a result, the structural engineer must also provide information about the units used in the mesh file. After a mesh model has been created, a related load model consisting of surface

distortion information can be generated. MODTool currently accepts NASTRAN .f06 files that contain the displacement of each node of the mesh. This information is used by the PO code, as described in Section 5.3 above. Figure 9 shows the Submit Data Mode window, with the load file 'load1.f06' used for the MIRO sub-millimeter wave case. Also shown is the associated mesh file, 'aug14\_2000.neu'. The user simply clicks on the 'Transfer the file to the server' button to complete the task.

#### 5.5 Geometrical Optics Mode

This mode allows the user to perform geometric optics ray-tracing calculations on the current project defined by its design and prescription. This is done by converting the design/prescription information into a MACOS input file, and running MACOS under the control of the GUI. The

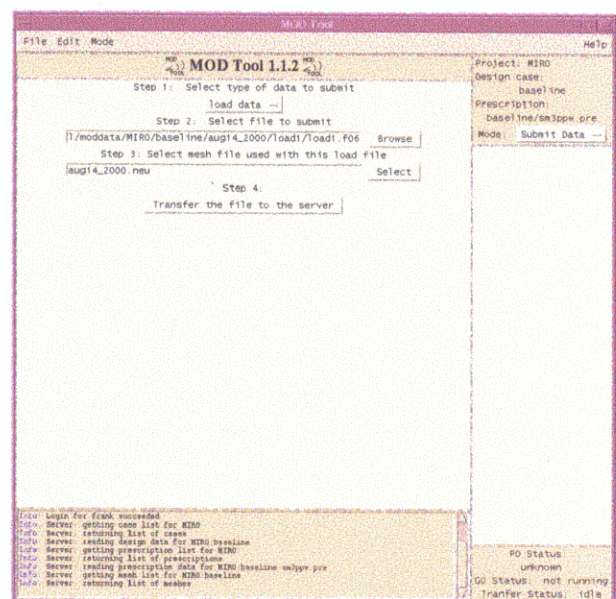


Figure 9. MODTool's Submit Data Mode window, showing mesh/load file combination submitted to the server to complete the analysis of the MIRO instrument under thermal loading conditions.

conversion is performed by the server, which also runs MACOS. Results are transferred to and displayed by the client. In contrast to the PO computation, this analysis is quite fast, and is usually used as a first-order estimate of the antenna system's performance. The Geometrical Optics Mode window is shown in Figure 2, and discussed in Section 4.

Once the designer has analyzed the antenna system in both its ideal and true environments, a direct comparison can be drawn. This information is typically conveyed to the project scientists, who then determine the proper course of action. For example, Figure 10 illustrates the E-plane directivity patterns for the MIRO instrument. Patterns corresponding



to both the nominal and distorted surfaces are shown. In this particular case, the relevant surface distortions were those induced not by thermal loading on the instrument along its trajectory, but by imperfections introduced during the mirror manufacturing process itself. Project scientists then decided, based on this same data, that the surface imperfections could indeed be tolerated.

## 6. DISCUSSION

This paper reviewed the MODTool software tool used for design, analysis and simulation of millimeter and microwave instruments. MODTool is a specific environment that allows collaborative design using high fidelity tools. Key to the development was the use of a common digital geometry that well represented the instrument geometry, and the use of high performance computers to decrease the turn-around time of the diffraction calculation.

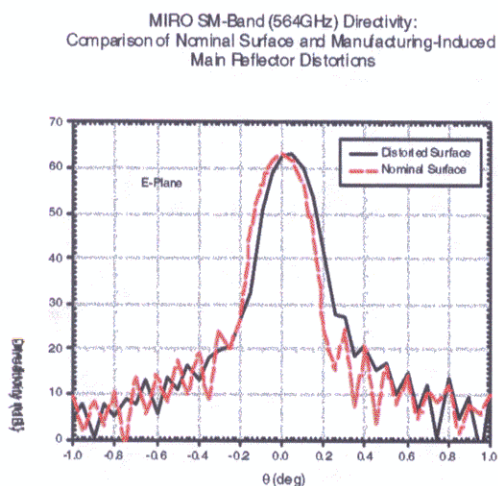


Figure 10. E-plane directivity patterns for the MIRO instrument with and without load.

Future extensions of MODTool would move toward integrating system level features into the code. For example, a signal-to-noise ratio could be extracted from the antenna pattern calculation and displayed in the collaborative environment. Other features like extended three-dimensional display of data could also be added.

## 7. ACKNOWLEDGMENT

A number of people contributed to the original development of MODTool. These include Andrea Borgioli, Chuigang Fu, William A. Imbriale, Vahraz Jamnejad, and Paul L. Springer. Integration into the Payload Development Team was instigated by Knut Oxnevad. The research described in this paper was carried out by the Jet Propulsion Laboratory,

California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

## REFERENCES

- [1] K. Oxnevad, "A Concurrent Design Environment for Designing Space Instruments, Proceedings of the 9th Thermal & Fluids Analysis Workshop, Ohio Aerospace Institute/NASA Lewis Research Center, Cleveland, Ohio, August 31-September 3, 1998.
- [2] D. S. Katz and T. Cwik, "Large-Scale, Low-Cost Parallel Computers Applied to Reflector Antenna Analysis," *USNC/URSI Nat. Rd. Sc. Mtg.*, Atlanta, GA, 1998.
- [3] R. E. Hodges and W. A. Imbriale, *Computer Program POMESH for Diffraction Analysis of Reflector Antennas*, Jet Propulsion Laboratory, California Institute of Technology, February, 1992.
- [4] *Modeling and Control of Optical Systems User Manual*, version 2.4.1, Jet Propulsion Laboratory, California Institute of Technology, April 1997.
- [5] "The Microwave Instrument for the Rosetta Orbiter", URL: <http://mirowww.jpl.nasa.gov/>.



Tom Cwik has been supervisor of the JPL High Performance Computing Group since October 1996. Prior to this position he was a Member of Technical Staff in the JPL Microwave Engineering Group, arriving at JPL in 1988. His work involves the use of high-performance computing for NASA/JPL engineering and science applications. This includes the development and use of integrated design tools for instrument design at proposal and build stages; the invention and analysis of microdevice components for coupling and filtering in remote sensing instruments; and algorithm development for high performance computing applications.

Tom Cwik received his BS, MS, and Ph.D. degrees in Electrical Engineering from the University of Illinois, Urbana-Champaign, in 1979, 1981, and 1986, respectively. He has worked at the Very Large Array in Socorro, NM, the National Bureau of Standards in Boulder, CO, and as a Postdoctoral Fellow at the Norwegian Institute of Technology. In 1996 he was appointed an Affiliate Professor in the Department of Electrical Engineering, University of Washington, Seattle, WA. He has edited a book and journal volume, written 6 book chapters, published 25 refereed articles, 100 conference proceedings and has 1 patent pending.



**Daniel S. Katz** is the Applications Project Element Manager for the Remote Exploration and Experimentation (REE) Project at the Jet Propulsion Laboratory (JPL), California Institute of Technology, which he joined in 1996. Previous roles at JPL include: Senior Member of the

Information Systems and Computer Sciences Staff in the Advanced Computing Discipline, Team Leader for MODTool, and Contact Scientist for a grand challenge team working in geophysics in NASA's Earth and Space Science (ESS) project. From 1993 to 1996 he was employed by Cray Research (and later by Silicon Graphics) as a Computational Scientist on-site at JPL and Caltech, specializing in parallel implementation of computational electromagnetic algorithms.

His research interests include numerical methods and algorithms applied to parallel computing, and computational methods in both electromagnetic wave propagation and geophysics. He received his B.S., M.S., and Ph.D degrees in Electrical Engineering from Northwestern University, Evanston, Illinois, in 1988, 1990, and 1994, respectively. His work is documented in numerous book chapters, journal and conference publications, and NASA Tech Briefs. He is a senior member of the IEEE, webmaster for the IEEE Antenna and Propagation Society, and serves on the IEEE Task Force on Cluster Computing's Advisory Committee.



**Frank J. Villegas** (S'92-M'95-S'97) was born in Habana, Cuba in December 1964. He received the BSEE and MSEE degrees from the University of Houston, Houston, TX in 1993 and 1995 respectively. Currently, he is pursuing doctoral research in

numerical electromagnetics in the electrical engineering department at the University of California in Los Angeles.

From 1993-1995, he was a research assistant in the Dept. of Electrical and Computer Engineering at the University of Houston. From 1995-1998, he was with TRW ES&TD in Redondo Beach, CA, engaged in the design of passive microwave, millimeter-wave and MMIC components, and MMIC packaging issues. From 1998-1999 he was with WaveBand Corp. in Torrance, CA, working on R&D of millimeter-wave scanning antennas for military and commercial applications. Since 1999 he has been with the High Performance Computing Group at JPL in Pasadena, CA., where he has mainly been involved in various projects dealing with the application of supercomputing solutions to electromagnetic design problems.

His research interests include EM applications of periodic structures, traveling (leaky) wave and microstrip antennas, leakage phenomena in planar waveguide circuits, passive microwave and millimeter-wave components and MMIC design. He has written various journal and conference publications, and is currently a student member of IEEE.